Statistical Methods for Data Science

Mini Project 3 (Solution)

---

1. (a) The mean squared error of an estimator $\hat{\theta}$ of parameter θ can be computed using Monte Carlo simulation as follows: repeat the process of simulating a sample and computing $\hat{\theta}$ from the simulated sample a large number of times, and compute the average of the squared deviations between $\hat{\theta}$ and θ.

   (b) See `R` code for the calculation.

   (c) Figure 2 summarizes the MSEs of the ML and method of moment estimators of θ.

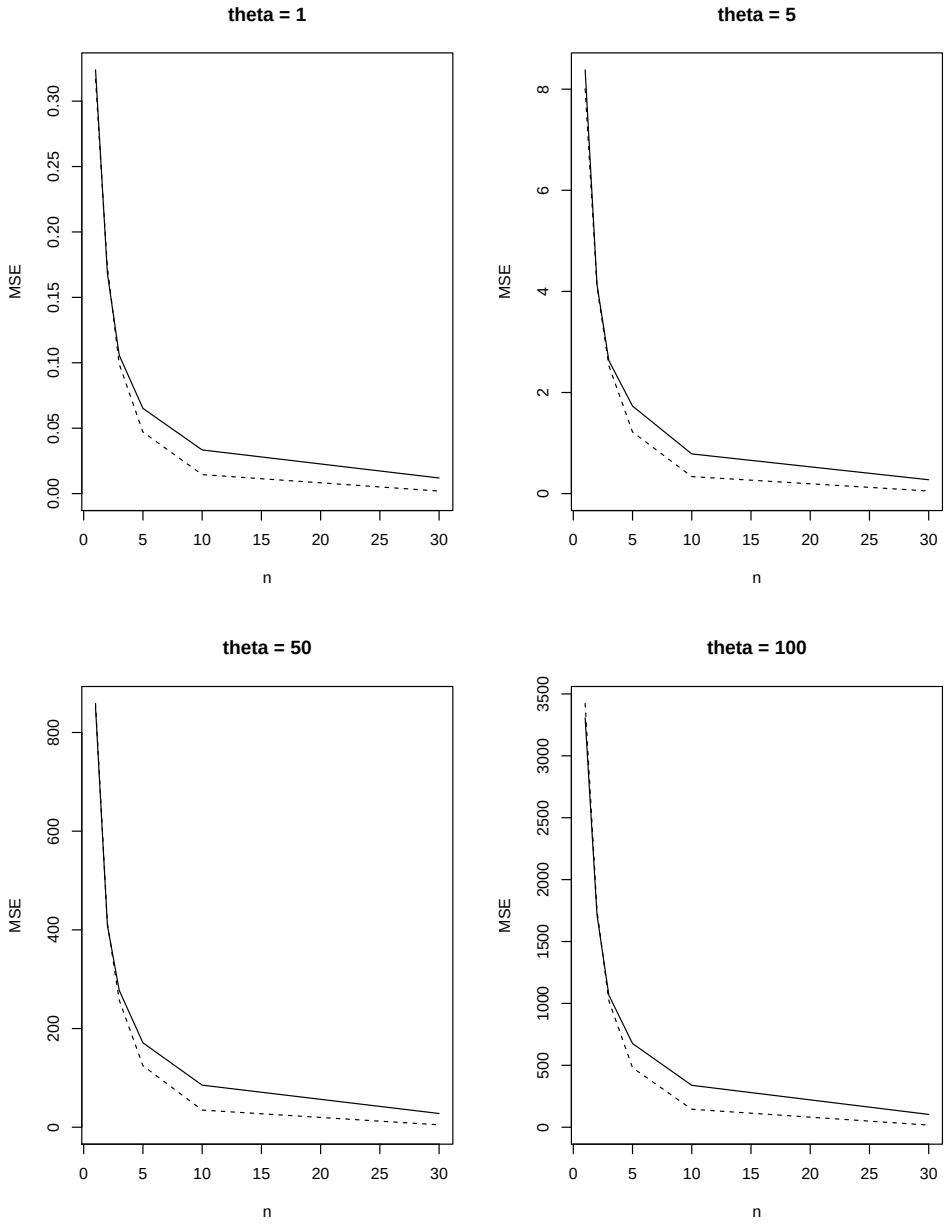

Figure 1: MSEs of the ML estimator (broken curve) and the method of moment estimator (solid curve) of θ as functions of n for various values of θ.

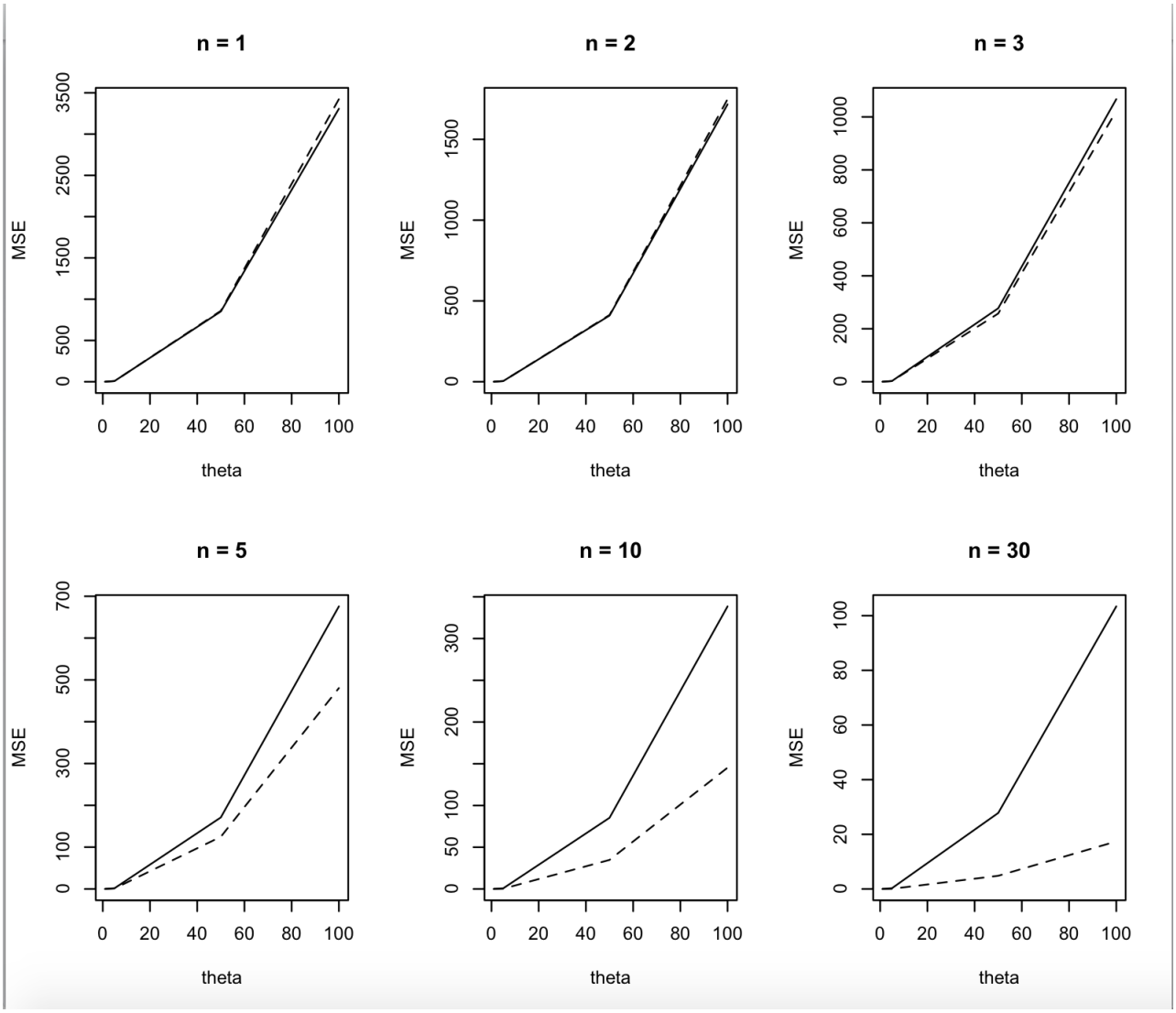

Figure 2: MSEs of the ML estimator (broken curve) and the method of moment estimator (solid curve) of θ as functions of θ for various values of n .

(d) The MSEs of both estimators decrease as n increases — this makes sense as the estimators become more accurate as n increases and in the limit they converge to the true parameter value (in which case the MSE would be zero). We also see that for $n = 1, 2$, the MSEs of the two estimators are practically the same. On the other hand, for $n \geq 5$, the MSE curve for the ML estimator ($\hat{\theta}_1$) lies below the MSE curve for the method of moment estimator ($\hat{\theta}_2$). These results hold for all the values of θ considered. Thus, we may conclude that for $n = 1, 2$, the two estimators are equally good, whereas for $n \geq 5$, the ML estimator is better. As a matter of fact, in this case, it is not hard to show that (try)

$$\text{MSE}(\hat{\theta}_1) = \frac{2\theta^2}{(n+1)(n+2)}, \quad \text{MSE}(\hat{\theta}_2) = \frac{\theta^2}{3n}.$$

From this result, we can deduce that the two estimators have the same MSE for $n = 1, 2$, and $\hat{\theta}_1$ has smaller MSE than $\hat{\theta}_2$ for $n > 2$, for all $\theta\,(> 0)$. Thus, we can theoretically show that the two estimators are equally good for $n = 1, 2$, and for $n > 2$, the ML estimator is better than the method of moment estimator.

2. (a) The probability density function is,

$$f(x) = \begin{cases} \frac{\theta}{x^{\theta+1}}, & \text{if } x \geq 1 \\ 0, & \text{if } x < 1 \end{cases} \tag{1}$$

where $\theta > 0$ is an unknown parameter.
Let $X_1, \ldots, X_n$ be a random sample of size n from this population.

The likelihood function,

$$L(\theta \mid \mathbf{x}) = \prod_{\mathbf{i=1}}^{\mathbf{n}} \frac{\theta}{\mathbf{x_i^{\theta+1}}}$$

The log likelihood function,

$$\begin{aligned} l(\theta \mid \mathbf{x}) &= log(\prod_{i=1}^{n} \frac{\theta}{x_i^{\theta+1}}) \\ &= \sum_{i=1}^{n} \{log(\theta) - (\theta+1)log(x_i)\} \\ &= nlog(\theta) - (\theta+1)\sum_{i=1}^{n} log(x_i) \end{aligned} \tag{2}$$

Taking the derivative of log likelihood function and setting equal to 0,

$$\begin{aligned} l^{'}(\theta \mid \mathbf{x}) &= \frac{n}{\theta} - \sum_{i=1}^{n} log(x_i) = 0 \\ \hat{\theta} &= \frac{n}{\sum_{i=1}^{n} log(x_i)} \end{aligned} \tag{3}$$

Taking the second derivative of log likelihood function,

$$l^{''}(\theta \mid \mathbf{x}) = -\frac{\mathbf{n}}{\boldsymbol{\theta^2}} < \mathbf{0}$$

Therefore, $\hat{\theta}_{mle} = \frac{n}{\sum_{i=1}^{n} log(x_i)}$.

(b)

$$\begin{aligned} \hat{\theta}_{mle} &= \frac{5}{\sum_{i=1}^{5} log(x_i)} \\ &= \frac{5}{log(21.72) + log(14.65) + log(50.42) + log(28.78) + log(11.23)} \\ &= 0.323 \end{aligned} \tag{4}$$

(c) $\hat{\theta}_{optim} = 0.323$. Yes, answers match.

(d) The estimated standard error and approximate 95% confidence interval for θ is 0.144 and [0.039, 0.606] respectively.
By large sample properties of MLE, $\hat{\theta}_{mle}$ follows asymptotically normal distribution if n is large. But in this case, the sample size $n(= 5)$ is very small. Therefore, these approximations cannot be considered as good.

R code:

```
##########################
# R code for Exercise 1  #
##########################

#################################

# function to compute MSE for a given (n, theta)

mse.fun <- function(n, theta, nsim = 1000) {
# function to simulate one sample and compute the two estimates
sim.fun <- function(n, theta) {
x <- runif(n, min = 0, max = theta)
mle <- max(x)
mome <- 2*mean(x)
return(c(mle = mle, mome = mome))
}
# replicate N times
est <- replicate(nsim, sim.fun(n, theta))
# compute MSE
return(rowMeans((est - theta)^2))
}

#################################

n <- c(1, 2, 3, 5, 10, 30)
theta <- c(1, 5, 50, 100)

# set the seed so that the results are reproducible

set.seed(12345)

mse.mle <- matrix(NA, nrow = length(n), ncol = length(theta))
mse.mome <- matrix(NA, nrow = length(n), ncol = length(theta))

for (i in 1:length(n)) {
for (j in 1:length(theta)) {
result <- mse.fun(n[i], theta[j])
mse.mle[i, j] <- result["mle"]
mse.mome[i, j] <- result["mome"]
}
}

#################################

# plot the results

par(mfrow = c(2, 2))

for (i in 1:length(theta)) {
```

```
plot(n, mse.mome[, i], type = "l", lty = 1,
ylim = c(0, max(mse.mome[, i], mse.mle[, i])),
main = paste0("theta = ", theta[i]),
ylab = "MSE")
lines(n, mse.mle[, i], lty = 2)
}

###################################

#########################
# R code for Exercise 2  #
#########################

values <- c(21.72, 14.65, 50.42, 28.78, 11.23)


# Negative of log-likelihood function
neg.loglik.fun <- function(theta, dat) {
result <- sum(log(theta) - ((theta + 1) * log(dat)))
return(-result)
}


# Minimize -log (L), i.e., maximize log (L)
ml.est <- optim(par = 0.3, fn = neg.loglik.fun, method = "L-BFGS-B",
lower = (10 ^ (-10)), hessian = TRUE, dat = values)

# standard errors and confidence intervals
sqrt(1/ml.est$hessian)
ml.est$par + c(-1, 1)* qnorm(0.975)*sqrt(1/ml.est$hessian)

###################################
```